

# Dynamic Report Generation in R: L<sup>A</sup>T<sub>E</sub>X vs. Markdown

Kate Hanley

Mango Solutions

11th March 2014

# What is knitr?

- “Elegant, flexible and fast dynamic report generation with R”
- An R package which allows R code and a document template to be combined
- Changes to the R code filter through to the document when the report is re-run

## Why use knitr?

**Fewer mistakes** - when updating your R code it is easy to forget to update your report at the last minute

**Reproducibility** - the code used to perform the analysis is linked to the report, so can be easily reproduced

**No copy/paste** - no need to copy your results and graphics into a word or powerpoint document

**Custom report styles** - can create templates for custom styles and layouts

# Why not?

**Learning curve** - you need to learn a mark-up language first

## How does it work?

- Start with your report template (more on this later!)
- Put your R code in a “chunk” in the document:

In latex a chunk looks like:

```
<< my chunk name, ... >>=
```

```
my R code
```

```
@
```

In markdown a chunk looks like:

```
```{ my chunk name, ... }
```

```
my R code
```

```
```
```

## How does it work?

- Load the knitr package
- Use the knit function to knit the document into the required format:

Latex:

```
library(knitr)
knit("myReport.Rnw")
```

Markdown:

```
library(knitr)
knit("myReport.Rmd")
```

**Fun fact:** Rnw stands for R Noweb

# What do I get?

## Latex

R Noweb file (.Rnw)



Latex file (.tex)



PDF

## Markdown

R Markdown file (.Rmd)



Markdown file (.md)



HTML

# “Hello World!” example

## Latex

```
\documentclass{article}
\begin{document}

Hello World!

\end{document}
```

## Markdown

Hello World!



# How to make a title

## Latex

```
\section{My Title}  
\subsection{My Subtitle}
```

## Markdown

```
%My Title  
%%My Subtitle
```

# How to make a list

## Latex

```
\begin{itemize}  
\item Item 1  
\item Item 2  
\end{itemize}
```

## Markdown

- Item 1
- Item 2

# How to make a table

## Latex

```
\begin{tabular}{|c|c|}  
 \hline  
   Column 1 & Column 2 \\  
 \hline  
   Value 1a & Value 2a \\  
   Value 1b & Value 2b \\  
 \hline  
 \end{tabular}
```

## Markdown

```
Column 1	Column 2
Value 1a | Value 2a  
Value 1b | Value 2b
```

## Latex vs Markdown

| Feature                          | Latex | Markdown |
|----------------------------------|-------|----------|
| Bullet points / numbered lists   | Yes   | Yes      |
| Auto-contents and page numbering | Yes   | No       |
| Dynamic figure referencing       | Yes   | No       |
| Customised styling               | Yes   | Yes      |
| Simple tables                    | Yes   | Yes      |
| Complex tables                   | Yes   | No       |

- NB** - you can use any HTML as part of a markdown document
- you can customise layout, create complex tables for markdown using CSS

# Latex

## Why?

- Flexible
- Can do almost anything you need to do in-house
- Mathematical typesetting is excellent
- Auto-generates a contents page and page/figure numbers
- Lots of packages available, and many help forums and guides

## Why not?

- Quite verbose
- Sometimes it can take ages to do something that seems very simple
- Learning curve

# Markdown

## Why?

- Very simple syntax
- Quick to learn and write

## Why not?

- Cannot create complex tables or dynamic references
- Need to know CSS to add styling to a document
- Need to know HTML to create anything complex
- Creates HTML document

# Which should I use?

It depends on your situation, but as a general rule:

- If you know latex, use latex
- If you want to do simple, quick documents without styling (or if you know CSS and HTML well), use markdown
- If you want to create complex documents with page numbers, contents, figure referencing and labelling, use latex

Thank you for listening!

Any questions?